EXHIBIT U

MAIC

SOFT COMPUTING

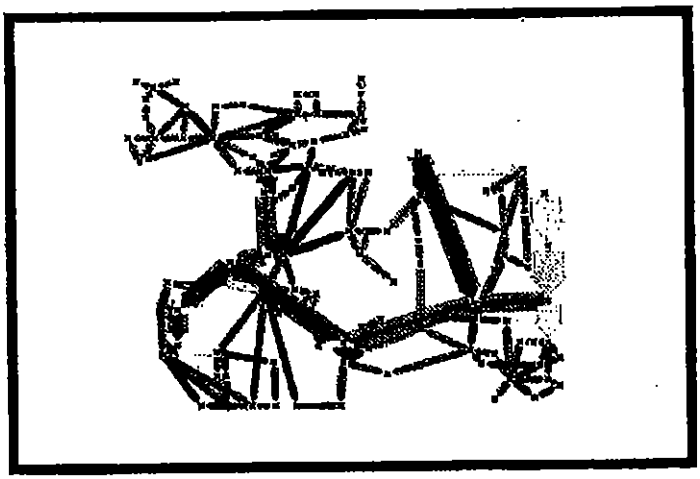
Rough Sets

Fuzzy Logic

Neural Networks

Uncertainty Management

Knowledge Discovery



Edited by
T. Y. Lin
and
A. M. Wildberger

SCS
SPONSORED BY
THE SOCIETY FOR COMPUTER SIMULATION
ISBN 1-56555-077-3

Statistical Methods for Computer Usage Anomaly Detection Using NIDES (Next-Generation Intrusion Detection Expert System)

Alfonso Valdes and Debra Anderson (SRI International) January 27, 1995

Abstract : With the dramatic growth of computer networks in recent months, the need to protect the integrity of information assets from unauthorized use has never been greater. SRI's Next Generation Intrusion Detection Expert System (NIDES), a fielded system for computer system monitoring, has both an expert system component and a statistical component integrated into a client-server model and communicating to a system security officer via a window-oriented user interface. The statistical component of NIDES, NIDES/STAT, is a nonparametric anomaly detection subsystem that has been successfully used to identify suspicious behavior on the part of computer users as well as UNIX applications. NIDES/STAT learns profiles of usage behavior and then flags deviations of short-term behavior from its more slowly varying historical profiles without knowing *a priori* scenarios of computer misuse and without constructing any sort of discriminant function between subjects. Results from recent studies indicate that NIDES/STAT is quite successful in detecting abnormal patterns in application usage data.

1 Introduction

With the growing concern over security of computer systems, several organizations have developed methods to automatically detect computer usage that is possibly improper or unauthorized. One such system, SRI's NIDES (Next Generation Intrusion Detection Expert System) [1], examines audit trail information using a custom nonparametric statistical component as well as a rulebased component, and is capable of processing audit records in real time or batch mode. The statistical component, NIDES/STAT, generates an anomaly score for each audit record by comparing recent observations with a long-term or historical profile which NIDES learns for each subject. Recent observations are maintained in a short-term profile, which is an integrated summary of a subject's activity typically spanning one to a few hundred audit records (intended to reflect minutes on a typical UNIX system).

The NIDES paradigm models a computer system as a set of subjects who initiate actions that affect objects. In this paradigm explored by SRI [1, 2, 5, 6], subjects can include computer users, applications, processes, network hosts, and so forth. To date, we have successfully employed NIDES to detect anomalous use with both computer users and application programs as subjects. Herein, we present an overview of the NIDES/STAT methodology and the results of an experiment profiling UNIX applications with NIDES.

2 NIDES Statistical Algorithm Description

The statistical approach used in NIDES [3] is to compare a subject's short-term behavior with its historical or long-term behavior, considering both long-term behavior absent from short-term behavior, and short-term behavior atypical of long-term behavior. Whenever short-term behavior is sufficiently unlike long-term behavior, a warning flag is raised. In general, short-term behavior is somewhat different from long-term behavior, because short-term behavior is more concentrated on specific activities and long-term behavior is distributed across many activities. To accommodate this expected deviation, the NIDES statistical component keeps track of the amount of deviation that it has seen in the past and issues a warning only if this deviation exceeds a subject-specific threshold.

The actual processing is as follows. The statistical component receives audit data either in real time or from files. This audit data serves both to score current behavior and to train profiles, with the former operation only possible after a period of initial profile training. As each audit record is received, NIDES modifies a fading memory summary of the recent activity with the activity observed on the newly arrived record. The fading memory concept is implemented by exponential aging of various summary counts. NIDES also maintains summaries of all the activity since the last long-term profile update. The long-term profile is updated once per day in real-time operation, or when the audit data stream timestamps cross a date boundary in batch processing. Updating consists of exponential fading of the existing long-term profile and combining the summary of activity maintained since the last update.

The NIDES statistical approach requires no *a priori* knowledge about what type of behavior would result in compromised security. It simply compares short-term and long-term behaviors to determine whether they are statistically similar. This feature of the NIDES statistical component makes it ideally suited for the task of computer usage anomaly detection in the absence of intrusion scenarios.

2.1 Profile Training

NIDES describes subject behavior by means of a profile, which we separate into short-term and long-term components. Anomaly scoring compares counts in a short-term profile with expected counts (which are based on historical probabilities maintained in a long-term profile) by means of a normalized square difference measure that is computationally chi-square like in form. Aspects of subject behavior are represented as measures (characterizations of usage along such dimensions as file access, CPU usage, hour of use, and so forth). The observed difference is compared on a measure per measure basis with the empirical distribution of the historically observed difference, from which we obtain a half-normal deviate that is now comparable across all measures. The squares of these are summed and compared to historically determined thresholds.

NIDES profile training consists of three phases: NIDES first learns the historical probabilities with which various categories are observed, then NIDES learns the empirical distribution of the deviation of short-term observations about these long-term category distributions, and finally NIDES sets the score thresholds. The short-term profile changes every audit record, while the long-term profile changes at regularly scheduled profile updates. Exponential fading of both profiles (using different time constants) permits relatively compact representation of the profiles as well as a mechanism for forgetting activity from the distant past not repeated in the recent past. Exponential fading of the short-term profile takes place as each audit record is received, while fading of the long-term profile takes place at profile update time. As a consequence of this aging mechanism, we speak of effective counts when referring to a count (for example, the number of times a category has been observed) to which this aging has been applied.

NIDES uses four classes of measures: activity intensity, audit record distribution, categorical, and continuous. The activity intensity measures determine whether the volume of activity generated is normal. The audit record distribution measure determines whether, for recently observed activity, the types of actions being generated are normal. The categorical and continuous measures examine whether, within a type of activity (say, accessing a file), the types of observed categories are typical. Categorical measures are those for which the observed values are by nature categorical. For example, the file access measure would have as its categories the names of accessed files. Continuous measures are those for which the observed values are numeric, such as CPU usage. For each measure, we construct a probability distribution of short-term and long-term behaviors. For example, for the measure of file access, the long-term probability distribution would consist of the historical probabilities with which different files have been accessed, and the short-term probability distribution would consist of the recent probabilities with which different files have been accessed. In this case, the categories to which probabilities are attached are the file names. In the case of continuous measures, such as CPU time, the categories to which probabilities are attached are ranges of values, which we refer to as bins (these ranges of values are mutually exclusive and span all values from 0 to infinity). The bins are scaled multiplicatively so that the range of values is logarithmically assigned with the ratio of the first to the last bin endpoint being approximately 1000. For example, if 10 bins are available, the right endpoint of the 10th is chosen to be the data mean plus three to four standard deviations, and then each bin endpoint proceeding downward is obtained by halving the next higher. This gives a factor of ten powers of two (or 1024), as desired. The binning procedure uses fractional powers of two for any number of bins other than ten. This mechanism is sufficiently robust that the scaling parameter need not be precisely estimated. By the application of this procedure, NIDES transforms continuous measures to categorical from the standpoint of its internal computations.

Category counts and probabilities are maintained as follows. For each measure, we define the following parameters:

- H_{Neff}: historical effective n
- P_i: historical probability, category i
- $Count_i$: arithmetic count since the last long-term profile update of the number of observations of category i
- $Agecount_i$: count of observations of category i since last update, with fading
- $NCATS$: number of categories for the measure

In addition, we define the following global parameters:

- γ_s: short-term fading factor
- γ_l: long-term fading factor

We then describe the processing for each measure. Suppose that on the current audit record, category i is observed for the measure of interest. The aged and unaged counts in the short-term profile are adjusted by aging the existing counts of all categories for the measure (using the short-term aging factor) and incrementing the

count of the observed category. Algorithmically, this consists of the following steps:

$$Count_i = Count_i + 1 \ ,$$

$$Agecount_j = \gamma_s \times Agecount_j, j \neq i \ ,$$

$$Agecount_i = \gamma_s \times Agecount_i + 1 \ .$$

The *Count* field is used to modify the long-term profile at the next update interval as follows. At each update time, the counts of observations since the last long-term profile update across all categories for a given measure are totaled into "today's count" (*TodayCount* below). The historical effective n (H_{Neff}) is aged by multiplying with the long-term aging factor γ_l. The contents of the long-term profile are converted from probabilities to effective counts by multiplying each historical bin probability P_i by H_{Neff}. The counts are then combined with the counts accumulated since the last update time $Count_i$, and converted back to probabilities (dividing by the new aged count). The algorithm for the updating is outlined below:

$$TodayCount = \sum_{i=1}^{NCATS} Count_i \ ,$$

$$H_{Neff} = \gamma_l * H_{Neff} \ ,$$

$$TempCount_i = H_{Neff} \times P_i + Count_i \ ,$$

After doing the above for all categories i, the historical count is updated and the totals converted to probabilities:

$$H_{Neff} = H_{Neff} + TodayCount \ ,$$

$$P_i = TempCount_i / H_{Neff} \ .$$

After these calculations, the category probabilities are examined to see if they should be dropped or grouped into a rare class. NIDES has mechanisms for dropping categories that fall below some threshold probability and grouping as rare those categories above the drop threshold but with sufficiently small probabilities that they might affect the statistical stability of the algorithms.

2.2 Differences Between Long- and Short-term Profiles — The Q Statistic

The *Agecount* field is used for estimating the difference statistic for each audit record. The degree of difference between the long-term profile for a measure and the short-term profile for a measure is quantified using a chi-square-like statistic, with the long-term profile playing the role of the actual probability distribution and the short-term profile playing the role of the observations. We call the resultant numerical value Q; there is a different Q value for each measure, updated as each audit record is encountered. Large values of Q mean that the long-term and short-term profiles for the measure are very different from one another and therefore warrant some suspicion; a Q value of zero means that they agree exactly.

We let N_{eff} denote the short-term effective n, mathematically defined as

$$N_{eff} = \sum_{i=0}^{Nobs} \gamma_s^i \ .$$

This quantity has an asymptotic value of $\frac{1}{1-\gamma_s}$, although the algorithm uses the actual value given by the above expression. The calculation of the Q statistic proceeds as follows:

$$e_i = N_{eff} \times P_i \ ,$$

$$Q = \sum_{i=1}^{NCats} \frac{(e_i - AgeCount_i)^2}{e_i} \ .$$

The reader may note that Q is similar to a chi-square random variable. Unfortunately, it is not possible to refer Q directly to a chi-square table due to potential dependence and insufficient observations for some bins in the data stream on which Q is based. Since the distribution of Q is not chi-squared, we need to track its values to determine what its distribution looks like. We observe the values for Q (computed on each audit record) and build an empirical probability distribution for Q using an aging and updating mechanism similar to that used for the measure categories. There is a Q statistic and a corresponding Q distribution for each measure. The Q distributions look somewhat like long-tailed and stretched-out chi-square distributions. Let QP_j be the empirical probability that Q falls in bin j of its distribution, and let TP_j be the corresponding tail probability, obtained as

$$TP_j = \sum_{k=j}^{NBins} QP_k \ .$$

2.3 Scoring Anomalous Behavior – The S and $T2$ Statistic

We transform the tail probability for Q and denote the transformed variable as S, defining the transformation so that S has a half-normal distribution. (A half-normal distribution looks like the right-hand side of a normal

distribution, except that the height of the probability distribution is doubled so that there is still unit area under the curve. This is also the distribution of the absolute value of a normally distributed variable.) The mapping from tail probabilities of the Q distribution to half-normal values is obtained by interpolation from a table of the tail of a normal distribution. Mathematically, the mapping from a tail probability to an S value takes the form

$$S = \Phi^{-1}(1 - TP/2).$$

As each audit record is received, we observe the bin in the Q distribution into which the computed value of Q falls, extract the corresponding tail probability value, and generate the corresponding S value according to the above equation. This is repeated for all measures, resulting in a vector of S values. High S values correspond to measures that are unusual relative to the typical amount of discrepancy that occurs between long-term and short-term profiles. Small S values correspond to measures that are not unusual relative to the amount of discrepancy that typically occurs between long-term and short-term profiles. We combine the S scores into an overall statistic that we call $T2$. This statistic is a summary judgment of the abnormality of all active measures, and is given by the sum of the squares of the S statistics normalized by the number of measures:

$$T2 = \frac{\sum_{m=1}^{Nmeas} S_m^2}{Nmeas}.$$

As is the case with Q, we build a long-term distribution for $T2$ rather than rely on a parametric model to obtain threshold values. The long-term distribution for $T2$ is built during the last stage of the profile building period, after reasonably stable long-term distributions for the Q statistics have been constructed. We declare recent audit records to be anomalous at the yellow or warning level whenever the $T2$ value is over the 1% threshold value of the long-term empirical distribution for $T2$, and at the red or critical level whenever the $T2$ is over the 0.1% threshold.

3 Detection of Anomalous Behavior in Application Usage

SRI adapted NIDES/STAT to detect masquerading applications from exit records extracted from UNIX audit data [4]. We examined approximately three months of application use in a UNIX environment, spiked with records from two applications representing abnormal usage, with four exit records for one application and one for another. We attempted to detect these records against the trained profiles from 26 legitimate applications. We observed 101 detections in 130 opportunities, corresponding to a detection rate of approximately 77%. This is the detection rate for a single execution of a masquerading program; the probability of eventually detecting masquerader activity from several executions of the program is much higher. For example, with this detection rate, the probability of detecting at least one of two executions of a masquerading program is approximately 95%. In addition to detection performance, any system such as NIDES must also achieve an acceptably low false positive rate, defined as the percentage of detections for a subject processed through its own profile. The observed false positive rate for legitimate applications for this experiment was 1.3%, based on observations not used in the training set (the nominal false positive rate was configured to be 1%). Table 1 gives the false positive results for our experiment, as well as a summary of the detection results for the masquerader applications.

3.1 Cross-profiling Experiment

Cross profiling is the term we use when running one subject's audit data through another subject's long-term profile. In such an experiment, we use the terms host to denote the application whose profile is being used and guest to denote the application supplying the data. Cross profiling allows us to determine how unique an application's profile is and how successful other applications might be in trying to masquerade as the host application. By examining the detection rate from cross profiling we can also assess the similarity of profiles among subjects with related functionality, with an eye to constructing group profiles. Grouping may be used to provide default initial profiles for subjects based on their group membership, allowing for a faster "bootstrapping" of the NIDES profile training mechanism.

Comparing the detection rates between applications shows three possible relationships: asymmetric detection, where an application can pass through the profile of another application (a low detection rate), but the other application cannot easily pass through the profile of the original application; mutually low detection, where pairs or small groups of applications can mutually pass through each other's profiles; and mutual detection, where for a pair of applications neither can pass through the other's profile without raising suspicion.

Table 2 shows the minimum, maximum, and average detection rates using the yellow threshold (1%) for each application. Subjects with high average detection rates, such as `getfullnm` and `latex`, are very sensitive

Application Profiling Results			
	False Positive		
	Yellow	Red	Detections
as	0.0	0.0	++-+ *
cat	3.9	1.9	-** *
compile	0.0	0.0	-+* *
cp	0.0	0.0	—* *
csh	0.5	0.5	+++* *
discuss	0.7	0.0	**++ *
emacs	2.0	0.3	**++ +
finger	0.0	0.0	-+* *
fmt	0.3	0.0	+*** *
gawk	1.3	0.0	++++ *
getfullnm	2.6	1.3	**** *
ghostview	0.9	0.0	**++ *
grep	0.1	0.0	—+ *
latex	3.9	0.0	**** *
less	0.7	0.4	+++* *
ls	1.0	0.1	—* *
mail	0.0	0.0	++++ *
make	2.2	0.0	+—+ *
man	0.9	0.0	-+* *
more	0.7	0.0	-++* *
mymoreproc	0.8	0.0	+*** *
pwd	0.4	0.4	**** *
rm	0.3	0.0	-+* *
sort	1.1	0.0	-++ *
stty	0.0	0.0	++** *
vi	1.3	0.1	— *
Total *			62
Total +			39
Total -			29

This table summarizes the false-positive and detection results for the application profiling study. The false positive column shows two percents: the percent of observations above the yellow (nominally 1%) detection threshold, and the percent above the red (nominally 0.1%) threshold. The column labeled "Detections" gives the result of processing each masquerading record through the host application's profile. We have recorded an asterisk (*) for detection above the critical (red) threshold, a plus (+) for detection above the warning (yellow) threshold, and a dash (-) for no detection. The groupings indicate the results for the four instances of the first masquerader followed by the single instance of the second. At the bottom of the table are total counts of the number of red (*), yellow (+), and non-detections (-).

Table 1: Application Profiling Results

to potential masquerader data. Others, with low average rates, such as `vi` and `grep`, are more tolerant and would be candidates for a masquerading attempt. The detection thresholds for `getfullnm` and `latex` are somewhat low, while those for `vi` and `grep` are on the high side. This may explain the low detection rate for masquerader data using `vi` as a host profile and confirms our low detection rates for `vi` under our true-positive tests in the first three experiments.

	Detection Percentages(%)		
Application	Minimum	Maximum	Average
as	0.10	100	53.62
cat	0.00	98.84	52.13
compile	0.90	98.33	24.71
cp	0.00	98.77	24.07
csh	0.00	99.17	42.76
discuss	7.04	99.49	72.87
emacs	10.53	98.80	86.23
finger	3.37	99.91	59.23
fmt	34.82	100.00	90.02
gawk	14.76	100.00	75.15
getfullnm	71.54	99.98	98.24
ghostview	5.92	99.31	82.21
grep	0.00	90.94	13.38
latex	94.62	99.81	98.53
less	1.02	99.69	48.04
ls	0.00	87.87	32.84
mail	1.15	99.68	64.27
make	1.70	96.83	51.27
man	5.88	99.96	63.32
more	0.16	86.85	35.06
mymoreproc	3.29	100.00	82.40
pwd	29.64	99.74	91.82
rm	0.00	97.01	34.81
sort	6.84	99.32	82.96
stty	49.59	99.76	95.43
vi	0.00	36.97	6.19

This table shows the minimum, maximum, and average detection percents for all applications processed through the profile of the host (row) application. For example, across all subjects, the average detection rate of the as profile was 53.62%.

Table 2: Detection Results for Cross-Profiling Experiment

4 Conclusions

We have presented a summary of the NIDES statistical methodology (NIDES/STAT) and the result of using this methodology to profile UNIX applications. NIDES/STAT is embedded in SRI's NIDES, an inte-

grated system for computer anomaly detection incorporating NIDES/STAT, a rule-based component, and a graphic user interface. NIDES/STAT does not rely on parametric models or some inter-subject distance function. It learns subject behavior by observing this behavior over time, and scores new behavior according to its similarity to past behavior. The methodology does not depend on any models of inappropriate computer use. Based on recent experimental results using actual UNIX audit data, NIDES/STAT is a powerful detector, correctly classifying 77% of records representing inappropriate usage while experiencing a false positive rate of 1.3%. It is evident that NIDES/STAT can successfully detect such records as well as distinguish between legitimate subjects. The proven performance of NIDES establishes it as a leading tool for those wishing to ensure the integrity of computer systems.

Acknowledgments:

Our research was supported by the U.S. Navy who funded SRI under contract N00039-92-C-0015 and by Trusted Information Systems through contract F30602-91-C-0067 which was funded by the U.S. Air Force, Rome Laboratory.

References

[1] D. Anderson, T. Frivold, A. Tamaru, A Valdes. NIDES User Manual/Computer System Operators Manual — Beta Release. Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, June 1994.

[2] R. Jagannathan, T. F: Lunt, F. Gilham, A. Tamaru, C. Jalali, P. Neumann, D. Anderson, T. D. Garvey, and J. Lowrance. Requirements Specification: Next Generation Intrusion Detection expert system(NIDES). Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, September 1992.

[3] H. S. Javitz and A. Valdes. The NIDES Statistical Component: Description and Justification. Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, March 1994.

[4] D. Anderson, T. Lunt, H. S. Javitz, A. Tamaru, A. Valdes. Detecting Unusual Program Behavior Using the NIDES Statistical Component. Technical Rreport, Computer Science Laboratory, SRI International, Menlo Park, California, December 1993.

[5] D. Anderson, T. Frivold, A. Tamaru, A. Valdes. Next Generation Intrusion Detection Expert System (NIDES) Software Design Specifications. Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, July 1994.

[6] T. F. Lunt, Ann Tamaru, Fred Gilham, R. Jagannathan, Caveh Jalali, H. S. Javitz, A. Valdes, P. G. Neumann, and T. D. Garvey. A Real-time Intrusion Detection Expert System (IDES), Final Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, February 1992.

CERTIFICATE OF SERVICE

I hereby certify that on the 30th day of June, 2006, I electronically filed the foregoing document, **DECLARATION OF PAUL S. GREWAL IN SUPPORT OF SYMANTEC CORPORATION'S OPPOSITION TO SRI INTERNATIONAL, INC.'S MOTION TO EXCLUDE FROM EVIDENCE THE EXPERT OPINION OF DANIEL TEAL, VOLUME 4 OF 4,** with the Clerk of the Court using CM/ECF which will send notification of such filing to the following:

John F. Horvath, Esq.
Fish & Richardson, P.C.
919 North Market Street, Suite 1100
Wilmington, DE 19801

Richard L. Horwitz, Esq.
David E. Moore, Esq.
Potter Anderson & Corroon LLP
Hercules Plaza
1313 North Market Street, 6th Floor
Wilmington, DE 19801

Additionally, I hereby certify that on the 30th day of June, 2006, the foregoing document was served via email and by Federal Express on the following non-registered participants:

Howard G. Pollack, Esq.
Michael J. Curley, Esq.
Fish & Richardson
500 Arguello Street, Suite 500
Redwood City, CA 94063
650.839.5070

Holmes Hawkins, III, Esq.
King & Spalding
191 Peachtree Street
Atlanta, GA 30303
404.572.4600

Theresa Moehlman, Esq.
King & Spalding LLP
1185 Avenue of the Americas
New York, NY 10036-4003
212.556.2100

/s/ Richard K. Herrmann
Richard K. Herrmann (#405)
Mary B. Matterer (#2696)
Morris, James, Hitchens & Williams LLP
222 Delaware Avenue, 10th Floor
Wilmington, DE 19801
(302) 888-6800
rherrmann@morrisjames.com

Counsel for Symantec Corporation